

(19) 日本国特許庁 (J P)

(12) 特 許 公 報 (B 2)

(11) 特許番号

第2503984号

(45) 発行日 平成 8 年(1996) 6 月 5 日

(24) 登録日 平成 8 年(1996) 4 月 2 日

(51) Int.Cl. <sup>8</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 9/38	3 7 0	8323-5E	G 0 6 F 9/38	3 7 0 B
7/00			7/00	A

発明の数 1 (全 7 頁)

(21) 出願番号 特願昭61-164748

(22) 出願日 昭和61年(1986) 7 月15日

(65) 公開番号 特開昭63-20536

(43) 公開日 昭和63年(1988) 1 月28日

(73) 特許権者 999999999

日本電気株式会社

東京都港区芝 5 丁目 7 番 1 号

(72) 発明者 松本 寛

東京都港区芝 5 丁目 33 番 1 号 日本電気株式会社内

(74) 代理人 弁理士 芦田 垣 (外 2 名)

審査官 久保 光宏

(54) 【発明の名称】 情報処理装置

1

(57) 【特許請求の範囲】

【請求項 1】パイプライン化した機能の異なる演算器を複数個持つ情報処理装置に於て、各演算に最低必要なパイプライン段数が異なる場合、各演算器に対してその後段に、単に結果の転送のみを行なうパイプライン用のレジスタを各演算系パイプラインの段数が等しくなるように必要な段数だけ付加し、前記付加したパイプライン用のレジスタの任意のレジスタからも演算結果を出力出来る切替手段と、各結果出力のうちから 1 つを選択しようとして同一演算器に於ける演算器出力及び付加したパイプライン用レジスタ出力の間で競合する場合には、パイプライン用レジスタのより後段からの出力を優先させることを基本とするが、そのなかのいずれかの結果出力をオペランドとして使う後続命令が待たされている場合には、該結果出力を、後段からの出力より優先させて出力

2

する制御手段とを含むことを特徴とする情報処理装置。

【発明の詳細な説明】

〔産業上の利用分野〕

本発明はパイプライン化した情報処理装置に関し、特に複数の命令を並列に実行するパイプライン化した情報処理装置に関する。

〔従来の技術〕

一般に、高性能を追求した情報処理装置に於ては、独立した各種演算器をパイプライン化し、同種の演算については、同じ演算器に対して連続的にオペランドを送り込んでパイプライン処理し、異なった演算については別の演算器に対して連続的にオペランドを送り込んで並列に処理することが行なわれている。また、各種演算器は、性能向上の為、それぞれ最も少ないパイプライン段数で実現している。

## 3

〔発明が解決しようとする問題点〕

ところが、同一演算器については、パイプライン処理の為、演算結果が入力順に順次出力されるが、異なった演算器からの結果の出力は、各演算器へのオペランドの投入タイミング、各演算器のパイプライン段数の違いにより、同一タイミングに競合する可能性がある。演算結果は一般にソフトウェア可視レジスタ群（以下、演算レジスタと呼ぶ）の中の一つに格納されるが、これは普通メモリ化されている為、同一タイミングに複数の結果を格納することはできない。

これを避ける為、各演算器のパイプラインの段数から逆算して、結果の出力タイミングが競合する場合には、各演算器へのオペランドの投入タイミング、即ち命令の実行起動を適当に遅らせている。命令の実行起動が、与えられた命令列の順にしか出来ない制御方式を採用している装置では、1命令の実行起動を遅らせることにより後続の命令も順次遅れることになり、性能の低下を招いている。

〔問題点を解決するための手段〕

本発明の情報処理装置はパイプライン化した機能の異なる演算器を複数個持つ情報処理装置において、各演算に最低必要なパイプライン段数が異なる場合、各演算器に対してその後段に単に結果の転送のみを行なうパイプライン用のレジスタを各演算系パイプラインの段数が等しくなるように必要な段数だけ付加し、前記付加したパイプライン用のレジスタの任意のレジスタからも演算結果を出力出来る切替手段と、全結果出力のうちから1つを選択しようとして同一演算器に於ける演算器出力及び付加したパイプライン用レジスタ出力の間で競合する場合には、パイプライン用レジスタのより後段からの出力を優先とさせることを基本とするが、そのなかのいずれかの結果出力をオペランドとして使う後続命令が待たされている場合には、該結果出力を、後段からの出力より優先させて出力する制御手段とを含んで構成される。

〔実施例〕

次に本発明について図面を参照して説明する。

第1図及び第2図は本発明の一実施例のブロック図及びその一部詳細図である。第1図に於て、乗算回路1、加減算回路2、論理演算回路3はそれぞれ各段が1マシンサイクルで動作するパイプライン化された演算器であり、1マシンサイクル毎に異なったオペランドの演算をパイプライン式に処理することが出来る。また各演算に必要なオペランドが入力オペランドバス1001、1002で与えられると、それぞれ7マシンサイクル、4マシンサイクル、2マシンサイクル後に演算結果を出力することが出来る。

加減算回路2及び論理演算回路3のパイプライン段数はそれぞれ4段、2段であり、乗算回路1のパイプライン段数は7段であるから、その差分、つまり3段及び5段分のパイプライン用レジスタ4～6、7～11が加減算回路2及び論理演算回路3に接続されている。

## 4

切替回路12は加減算回路2の出力、パイプライン用レジスタ4～6の出力のうち、いずれかを選択して出力する回路であり、切替回路13は論理演算回路3の出力、パイプライン用レジスタ7～11の出力のうち、いずれかを選択して出力する回路である。また切替回路14は乗算回路1、切替回路12、13の出力のうちいずれかを選択して、演算レジスタ16への書込みバス1003に出力する回路である。

本実施例には乗算回路1にはパイプライン用レジスタを設けていないが、これは説明を簡単にする為であり、加減算回路2、論理演算回路3と同様に後段にパイプライン用レジスタを適当な段数だけ設けても良い。この場合乗算回路に設けたパイプライン用レジスタの段数だけ、加減算回路、論理演算回路のパイプライン用レジスタの段数を共に増やすことになる。

制御回路15は切替回路12、13、14の選択を制御する回路である。まず切替回路14の選択制御は、各演算回路系の間で結果の出力の競合が発生したときにどの演算回路系からの出力を優先するかを決めるものであり、優先順は乗算回路系、加減算回路系、論理演算回路系の順である。これは例えば乗算回路にはパイプライン用レジスタが無いので結果が求まった時点ですぐに演算レジスタ16に転送しなければ結果が失われてしまうので最優先にするものであり、以下同様の考え方で演算に最低必要なパイプライン段数が多いものをより優先的にしている。

次に切替回路12の選択制御は、加減算回路系の結果の最大4ケースの出力の競合がある場合に、どれを優先するかを決めるものであり、特に結果を早く出力したいものが無い場合は、優先度はパイプラインのより後段のものを高くしている。これは例えば最後段のパイプライン用レジスタ6に結果がある場合は、これを優先して演算レジスタ16に転送しなければ、結果が失われてしまうので最優先にするものであり、以下同様の考え方で後段側をより優先度を高くしている。また、特に結果を早く出力したいものがある場合、例えばその結果をオペランドとして使う命令が実行待ちになっている場合にはすぐに結果を演算レジスタ16に格納し、読み出して実行に入らなければ性能の低下を招くので、これより後段側のパイプライン用レジスタに別の結果がまだ演算レジスタ16に転送されなくて残っている場合でも、差しつかえない限りこれを追い越して演算レジスタ16に格納するという制御を行なっている。切替回路13についても切替回路12と同様である。

フリップフロップ101～107、201～207、301～307はそれぞれ乗算回路系、加減算回路系、論理演算回路系の各パイプラインステージに有効データが存在することを表わす有効ビットすなわち有効フラグを表わすものであり、これと制御回路15の詳細を示すと第2図のようになっている。

第2図は主に加減算回路2に注目して詳細が記されて

## 5

いる。加減算回路2にオペランドが与えられると、パイプライン式に演算が進むにつれ、フリップフロップ201, 202, 203が順に“1”となり、演算結果が求まると同時にフリップフロップ204が“1”になる。この時に求まった結果を演算レジスタ16に転送できる場合は、転送が行なわれてしまうため、フリップフロップ204は“1”→“0”になり、フリップフロップ205は1にならずに終了する。他との競合があり優先度が低くて転送できない場合は、結果をパイプライン用レジスタ4に入れると同時にフリップフロップ204は“1”→“0”, フリップフロップ205は“0”→“1”になり、結果が演算レジスタ16に転送されずにパイプライン用レジスタ4に移ったことを示す。以下パイプライン用レジスタ5, 6, フリップフロップ206, 207も同様である。

格納アドレスレジスタ221~226は加減算結果を演算レジスタ16に格納する場合の格納アドレスを持ち回るレジスタであり、オペランドアドレスレジスタ231, 232はそれぞれ今から演算を開始しようとしている命令の2オペランドの演算レジスタアドレスを保持するレジスタで、現在実行中の命令の結果を待っている場合は実行に入れずこのレジスタに保持され続けるものである。

オペランドアドレスレジスタ231, 232の内容は格納アドレスレジスタ226, 225, 224と比較され、一致するものがあれば各パイプラインステージ対応に一致信号1601, 1602, 1603が“1”となる。これらの信号から優先度変更信号1701, 1702が作られる。有効ビット201~207に対応するパイプラインステージをA~Gと呼ぶと、この論理テーブルは次の様になっている。

Fステージ と一致(信 号1601)	Eステージ と一致(信 号1602)	Dステージ と一致(信 号1603)	優先度変更信号 (信号 1701)	優先度変更信号 (信号 1702)
0	0	0	1	1
0	0	1	0	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	1
1	0	1	1	0
1	1	0	1	1
1	1	1	1	1

つまりFステージにある演算結果よりD, Eステージにある結果が早く欲しい場合は、Fステージの有効ビットを抑える信号1701を“0”にし、Eステージにある演算結果よりDステージにある結果が早く欲しい場合はEステージの有効ビットを抑える信号1702を“0”にする論理となっている。

切替回路12の制御は、有効ビットをあらわすフリップフロップ204~207と優先度変更信号1701, 1702, 加減算回路系の結果を優先して出力できることを指示する信号1402によって行なわれる。先ず優先度変更信号1701, 1702が共に“1”の場合、つまり優先度変更が無い場合につい

## 6

てみると、信号1402が“1”であればフリップフロップ207, 206, 205, 204の順に見て例えば最初に“1”のあるフリップフロップが206であるとするこれに対応するパイプライン用レジスタ5の結果を演算レジスタ16に転送するため、ANDゲート217~214のうち対応するANDゲート216の出力信号1206が“1”となり、切替回路12ではパイプライン用レジスタ5が選択されて結果の転送が行なわれることになる。

信号1402が“0”であれば加減算回路系の結果は演算レジスタ16に転送できないのでANDゲート217~214のいずれの出力(1207~1204)も“0”となり、従って信号1306~1304は全て“1”となるのでフリップフロップ204~206の内容がそのままフリップフロップ205~207に移される。

次に優先度変更がある場合についてみると、例えばフリップフロップ207, 206, 205, 204のうち206と205が“1”であり、格納アドレスレジスタ226の内容とオペランドアドレスレジスタ231, 232とは一致せず(信号1601=“0”), 格納アドレスレジスタ225の内容とオペランドアドレスレジスタ231或は232が一致している(信号1602=“1”)とすると、次に実行に入ろうとしている命令はEステージにある演算結果を待っている状態であり、Fステージの結果よりも先にEステージの結果を優先して演算レジスタ16に格納した方が待合せ時間が少なくて済む。この場合、優先度変更信号1702=“0”となることによってフリップフロップ206の出力信号を抑え、ANDゲート216の出力信号は“0”, ANDゲート215の出力信号は“1”となって、切替回路12ではパイプライン用レジスタ4が選択され、Fステージにある結果よりEステージにある結果が転送されることになる。この場合追い越されて転送されなかったFステージ上の結果はGステージに移される。

なお、Gステージに有効なデータが格納されている場合には、最優先でGステージのデータが出力され、追い越しは中止される。これは、第2図において、Gステージの有効性を表すフリップフロップ207が“1”になると、優先度変更信号1701, 1702の如何にかかわらず、ANDゲート217, 216, 215, 214の出力信号1207, 1206, 1205, 1204は、それぞれ“1”, “0”, “0”, “0”となり、切替回路12ではGステージのデータが選ばれることで理解されよう。

次に切替回路14の選択制御について説明する。各演算回路系から演算レジスタ16に転送すべき結果が1つでも存在することを表わす信号1501, 1502, 1503はそれぞれフリップフロップ107の出力、フリップフロップ204~207の出力の論理和、フリップフロップ302~307の出力の論理和である。

この3本の信号を入力し、優先順位つまり乗算回路系、加減算回路系、論理演算系の順に従って、乗算回路系の出力を選ぶ信号1401, 加減算回路系の出力を選ぶ信

## 7

号1402, 論理演算系の出力を選ぶ信号1403を作成し, これによって切替回路14の選択制御を行なっている。

第3図は結果待ちを行なわない場合の高速化の一例を説明する為の命令列である。①の命令は演算レジスタ1の内容と, 演算レジスタ2の内容を乗算し, 結果を演算レジスタ3に格納することを意味する。②~⑦についても同様である。

第4-A図は, 第3図の命令列に於ける従来の装置, 即ち第1図に於けるパイプライン用レジスタ4~11, 切替回路12, 13が無い場合の命令実行のタイムチャートである。命令の実行起動がマシンサイクル毎に1命令で, 与えられた命令列順にのみ行なえる場合, ①~③の命令については1マシンサイクルずれて乗算回路1, 加減算回路2, 加減算回路2に演算を指示し, それぞれ7サイクル目, 4サイクル目, 4サイクル目に演算結果が求まるので演算レジスタ16への書き込みパス103には②③①の命令順に結果が得られる。結果の命令順が変わってもこの場合何ら問題はない。ところが④の命令については, タイミング4から実行開始させたとすると(第4-A図の点線), 加減算である為4マシンサイクル後のタイミング7で結果が求まり, ①の命令の結果が求まるタイミングと一致し, 不具合が生じる。従って④の命令の実行開始を1マシンサイクル遅らせ, タイミング5から実行させることによってこれを避けるようにする(第4-A図の実線)。⑤の命令についても同様である。こうして7命令を実行させると, 全体で16マシンサイクルを必要とする。

第4-B図は第3図の命令列に於ける本発明の装置, 即ち第1図でのタイムチャートである。この場合①~⑦の命令すべてについて, 前記の従来の場合とは異なり, 演算結果の競合を気にせず, 1マシンサイクルずれて実行開始させる。タイミング6で③と⑤の命令の結果が競合するが, 優先順に従って, ③の命令の結果が切替回路12, 14で選択され, 演算レジスタ16に転送される。⑤の命令の結果は演算レジスタ16には転送されず, パイプライン用レジスタ7に移される。次にタイミング7では①と④と⑤と⑥の命令の結果が競合するが, 優先順に従って①の命令の結果が切替回路14で選択され, 演算レジスタ16に転送される。④の命令の結果はパイプライン用レジスタ4に移され, ⑤, ⑥の命令の結果はそれぞれパイプライン用レジスタ8, 7に移される。次にタイミング8では④と⑤と⑥の命令の結果が競合するが, 優先順に従って④の命令の結果がパイプライン用レジスタ4から切替回路12, 14を通して演算レジスタ16に転送される。⑤と⑥の命令の結果はそれぞれパイプライン用レジスタ9, 8に移される。次にタイミング9では⑤と⑥の命令の結果が競合するが, 優先順に従って⑤の命令の結果がパイプライン用レジスタ9から切替回路13, 14を通して演算レジスタ16に転送される。⑥の命令の結果はパイプライン用レジスタ9に移される。タイミング10では競合なく⑥

## 8

の命令の結果がパイプライン用レジスタ9から切替回路13, 14を通して演算レジスタ16に転送される。⑦の命令の結果はタイミング13で競合なく結果の格納が行なわれ, 結果的には7命令の実行に13マシンサイクルで済むので, 高速化されることが理解されよう。

尚, パイプラインの長さを同じ長さに合わせているのは, パイプラインの長さがすべて同じであれば1マシンサイクル毎の入力に対し必ず出力の競合なしに1マシンサイクル毎に出力を得ることが出来るためであり, 単に転送の為のパイプライン用レジスタの任意の出力から結果を取り出す構成にしているのは, 競合さえなければ出来るだけ早く, 演算レジスタを更新し, この結果をオペランドとして使用する命令が後続の命令の中にあれば早く実行に移らせるためである。

第5図は結果待ちを行なう場合の一例を説明する為の命令列である。記述方法は第3図と同様の為, 説明は省略する。

第6-A図は, 第5図の命令列に於けるタイムチャートを優先変更回路が無いと仮定した場合のタイムチャートであり, ④, ⑤, ⑥の命令は順次タイミング9, 10, 11で結果の転送が行なわれる。⑦の命令は⑥の命令の結果が演算レジスタ16へ転送され読み出されて(タイミング12)はじめてタイミング13から実行開始される。これに対して, 本発明の装置に於ける優先変更回路により, 第6-B図のタイムチャートに示すように, ④, ⑤, ⑥の命令の結果の転送は, ⑦の命令が⑥の命令の結果を待つ実行待ちとなっているので, ⑥の結果を早く出力するように動作する。つまり, 優先変更回路により, タイミング9でDステージのアドレス一致信号1603が“1”となって⑥の命令の結果を先に出力し, 続いて④, ⑤の順に結果出力されるようになる。これによって, ⑦の命令はタイミング11から実行開始され, 第6-A図に比べて高速化が計られているのが理解されよう。

## 【発明の効果】

以上説明したように, 本発明はパイプライン化した機能の異なる演算器を複数個持つ情報処理装置に於て, 各演算に最低必要なパイプライン段数が異なる場合, 各演算器に対してその後段に単に結果の転送のみを行なうパイプライン用のレジスタを必要な段数だけ付加することによって各演算系パイプラインの段数を等しくし, 付加したパイプライン用レジスタの任意のレジスタからも演算結果を出力出来るように構成し, 全結果出力のうちから1つを選択しようとして同一演算器に於ける演算器出力及び付加したパイプライン用レジスタ出力の間で競合する場合には, 結果を早く必要とするものがあればこれを優先し, さもなければパイプライン用レジスタのより後段からの出力を優先させて出力させることによって必要なマシンサイクルを減少し, 動作を高速化させる効果がある。

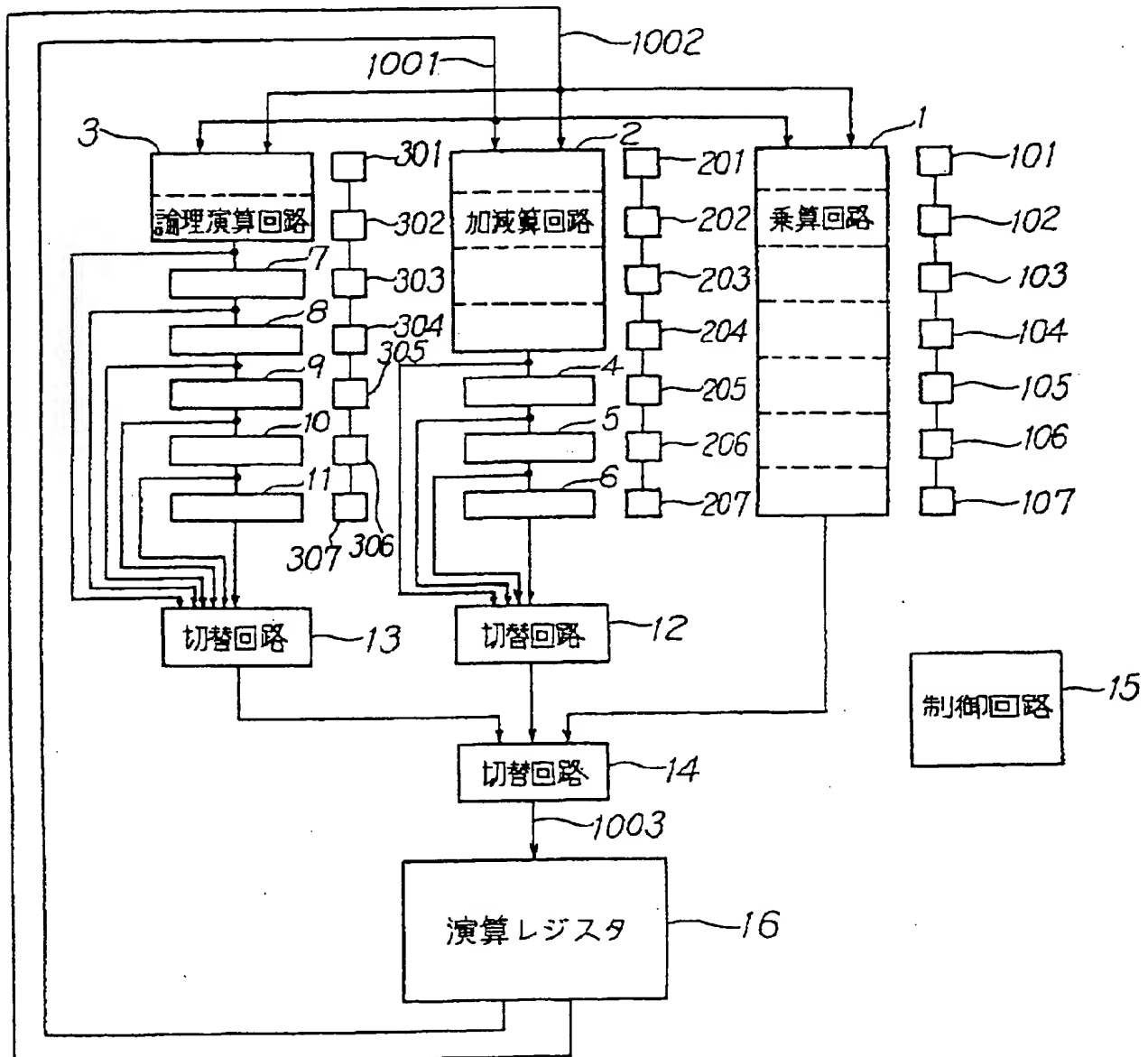
## 【図面の簡単な説明】

第1図は本発明の一実施例を示すブロック図、第2図は第1図の一部を詳細に示した図、第3図、第5図は本発明の効果を説明する為の命令列例、第4-A図、第4-B図、第6-A図、第6-B図は本発明の効果を説明する為のタイムチャートである。

記号の説明: 1……乗算回路, 2……加減算回路, 3……論理演算回路, 4~11……パイプライン用レジスタ, 12~14……切替回路, 15……制御回路, 16……演算レジスタ, 101~107, 201~207, 301~307……フリップフロップ (有効フ

ラグ), 214~217……AND回路, 221~223……結果アドレスレジスタ, 224~226……格納アドレスレジスタ, 231, 232……オペランドアドレスレジスタ, 241……比較器, 1001, 1002……オペランド供給バス, 1003……結果格納バス, 1204~1207……選択回路12に於ける選択信号, 1401~1403……選択回路14に於ける選択信号, 1501~1503……各演算系に格納すべき結果があることを示す制御信号, 1601~1603……結果アドレス/オペランドアドレス一致信号, 1701, 1702……優先度変更信号。

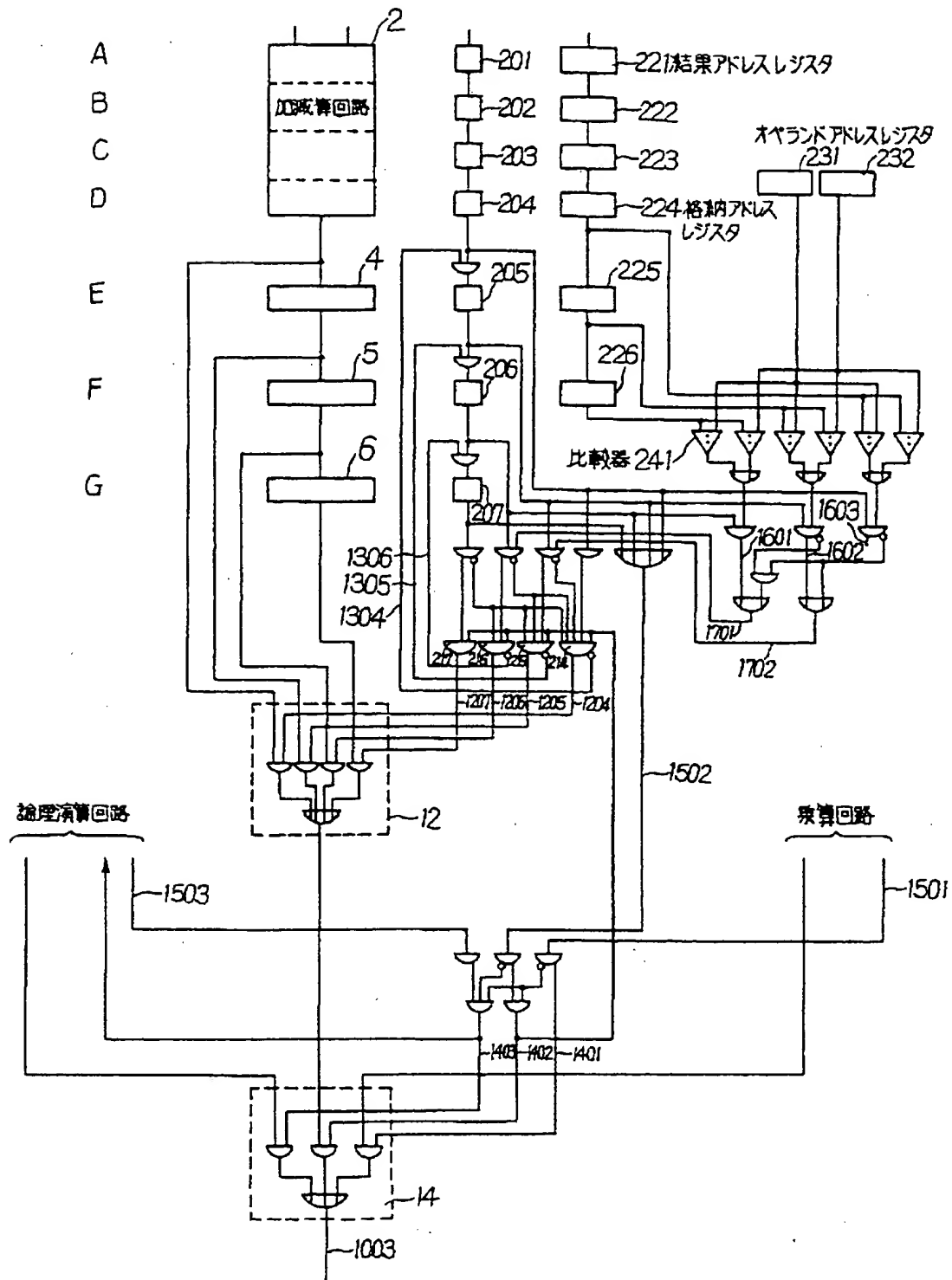
【第1図】



4~11: パイプライン用レジスタ

101~107, 201~207, 301~307: フリップフロップ (有効フラグ)

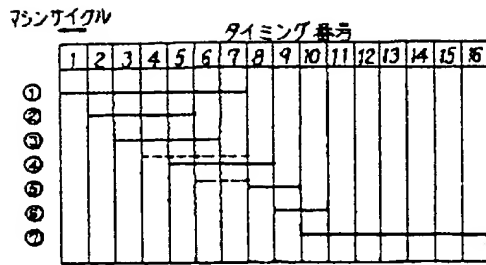
【第2図】



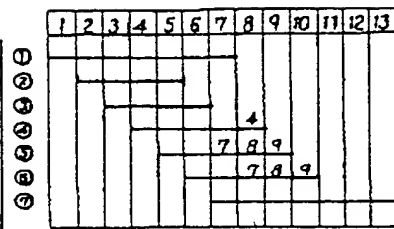
【第3図】

- ①  $R3 \leftarrow R1 \times R2$   
 ②  $R6 \leftarrow R4 + R5$   
 ③  $R9 \leftarrow R7 + R8$   
 ④  $R12 \leftarrow R10 + R11$   
 ⑤  $R15 \leftarrow R13 \cdot R14$   
 ⑥  $R18 \leftarrow R16 \cdot R17$   
 ⑦  $R21 \leftarrow R19 \cdot R20$

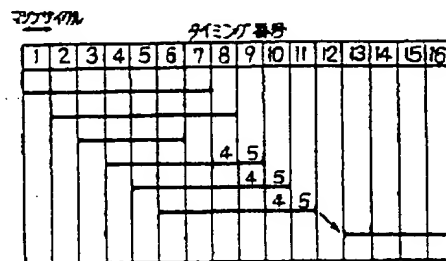
【第4-A図】



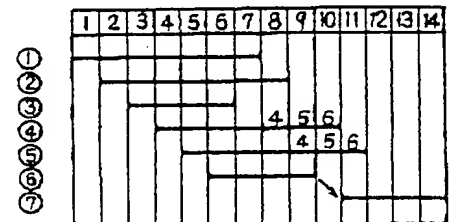
【第4-B図】



【第6-A図】



【第6-B図】



【第5図】

- ①  $R103 \leftarrow R101 \times R102$   
 ②  $R106 \leftarrow R104 \times R105$   
 ③  $R109 \leftarrow R107 + R108$   
 ④  $R112 \leftarrow R110 + R111$   
 ⑤  $R115 \leftarrow R113 + R114$   
 ⑥  $R118 \leftarrow R116 + R117$   
 ⑦  $R121 \leftarrow R118 + R120$

①  
②  
③  
④  
⑤  
⑥  
⑦